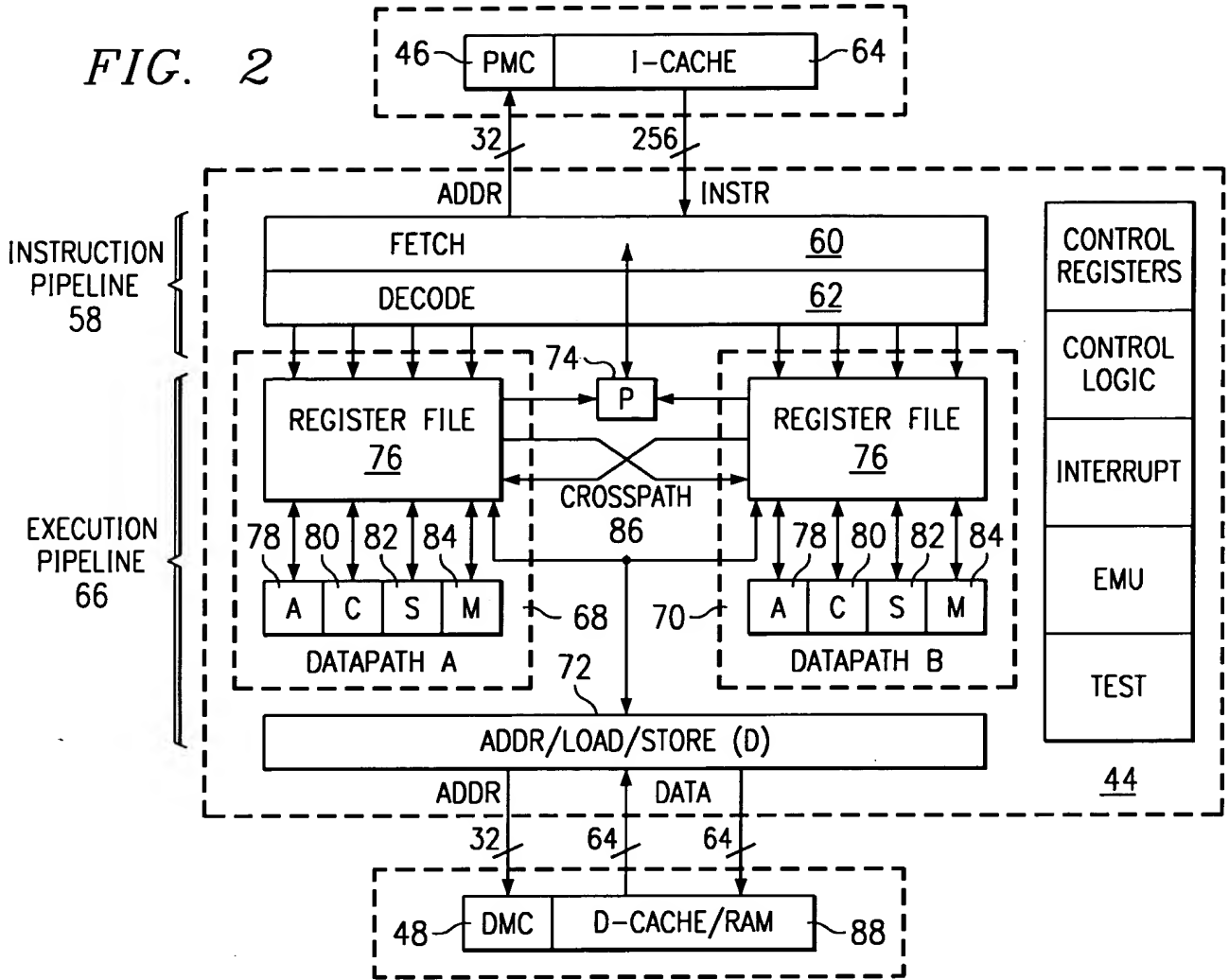


FIG. 2



UNIT GROUP	OPERATIONS	REGISTER FILE ACCESS	
		PRIMARY DATAPATH	ALTERNATIVE DATAPATH
A	GENERAL ARITHMETIC BOOLEAN AND CONTROL REGISTER ACCESS	R/W	R
C	COMPARE, SHIFT, BOOLEAN ARITHMETIC: ADD, SUB	R/W	R
S	SHIFT, ROTATE, EXTENDED BOOLEAN ARITHMETIC: ADD, SUB	R/W	R
M	MULTIPLY ARITHMETIC: ADD, SUB	R/W	R
D	LOAD STORE ADDRESS COMPUTATION	W TO BOTH R FROM BOTH R/W BOTH	
P	BRANCH	R FROM BOTH	

FIG. 3

R=READ, W=WRITE

09687540 101300

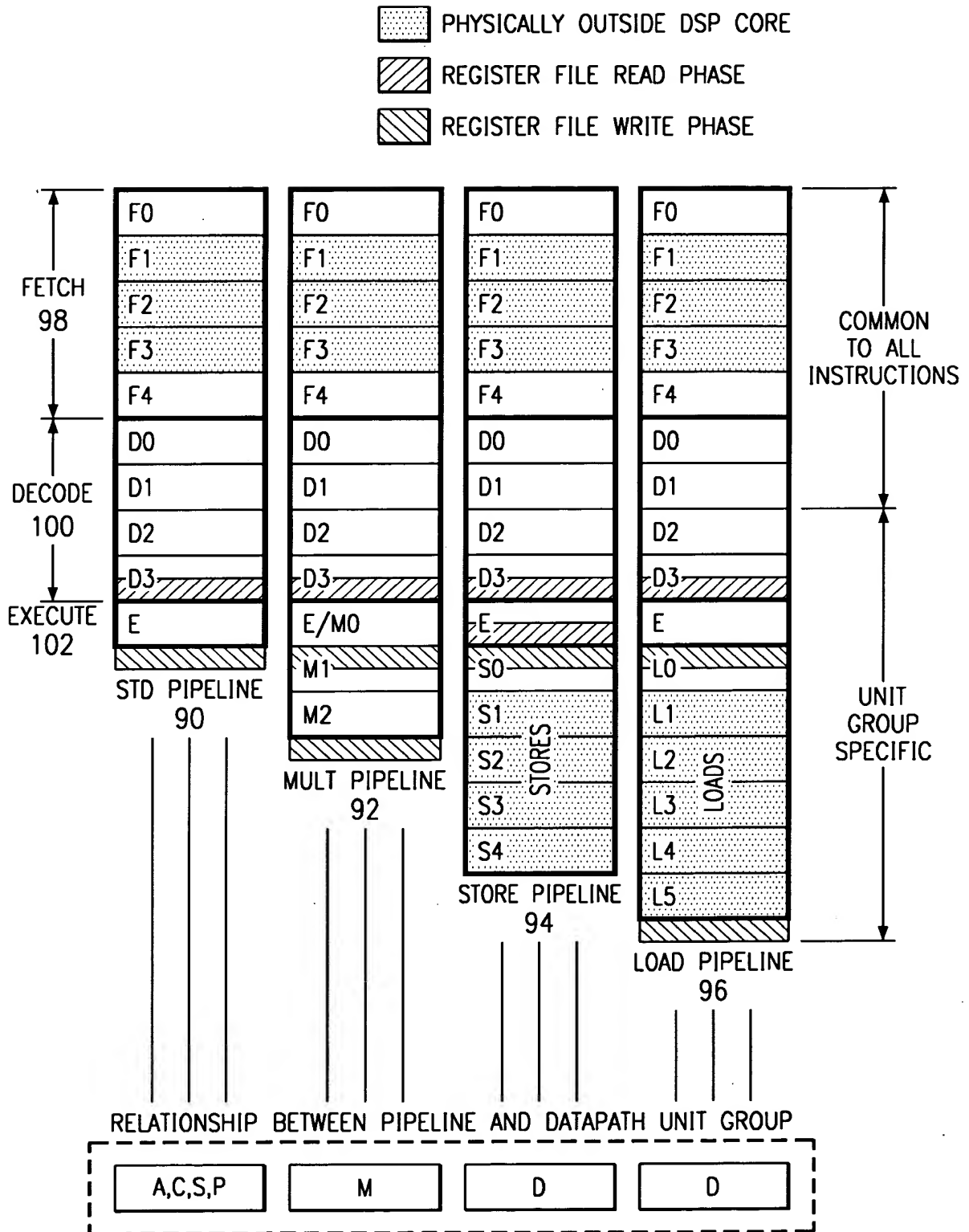


FIG. 4

STAGE	FUNCTION
F0	SEND PC TO PROGRAM MEMORY CONTROLLER. LDIP ASSIGNED.
F1	CACHE BLOCK SELECT.
F2	ADDRESS PHASE OF INSTRUCTION CACHE ACCESS.
F3	DATA PHASE OF INSTRUCTION CACHE ACCESS.
F4	FETCH PACKET SENT TO DSP.

FIG. 5a

STAGE	FUNCTION
D0	DETERMINE VALID INSTRUCTIONS IN CURRENT FETCH PACKET.
D1	SORTS INSTRUCTIONS IN EXECUTE PACKET ACCORDING TO DESTINATION UNITS.
D2	INSTRUCTIONS SENT TO DESTINATION UNITS. CROSSPATH REGISTER READS OCCUR.
D3	UNITS DECODE INSTRUCTIONS. REGISTER FILE READ (2ND PHASE).

FIG. 5b

UNIT	STAGE	FUNCTION
NON M UNIT	E	EXECUTION OF OPERATION BEGINS AND COMPLETES. FULL RESULT AVAILABLE AT END OF CYCLE.
M UNIT	M0	EXECUTION OF MULTIPLY OPERATION BEGINS. (OR, NON-MULTIPLY OPERATION BEGINS AND COMPLETES.)
M UNIT	M1	MULTIPLY OPERATION CONTINUES. (OR, NON-MULTIPLY RESULT WRITTEN TO REGISTER FILE (PHASE 1).)
M UNIT	M2	MULTIPLY OPERATION COMPLETES.

FIG. 5c

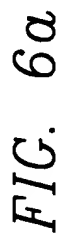
STAGE	FUNCTION
E	ADDRESS GENERATION OCCURS. REGISTER FILE ACCESS FOR READ DATA.
L0	LOAD ADDRESS GENERATED DURING E IS SENT TOWARDS THE DMC.
L1	ADDRESS DECODE, TC ARBITRATION, TAG COMPARES.
L2	ADDRESS DECODE, TC ARBITRATION, TAG COMPARES.
L3	ADDRESS PHASE OF DATA CACHE ACCESS.
L4	DATA PHASE OF DATA CACHE ACCESS.
L5	64-BIT DATA SENT TO DSP.

FIG. 5d

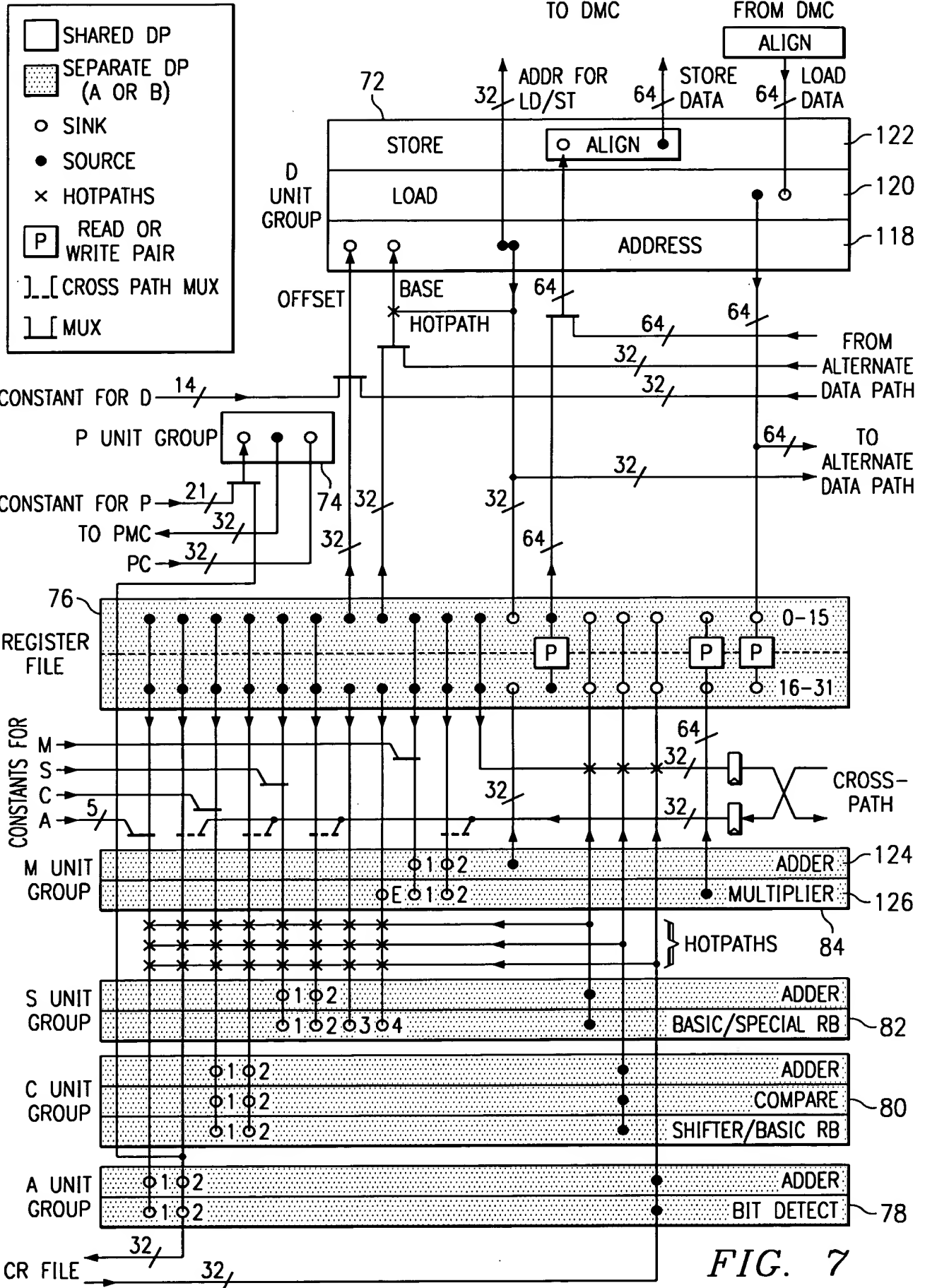
STAGE	FUNCTION
E	ADDRESS GENERATION OCCURS. REGISTER FILE ACCESS FOR WRITE DATA.
S0	ADDRESS SENT TO DMC.
S1	ADDRESS DECODE IN DMC. WRITE DATA ALIGNMENT.
S2	TAG COMPARE IN DMC. WRITE DATA SENT TO DMC.
S3	ADDRESS PHASE IN DATA CACHE.
S4	DATA PHASE IN DATA CACHE.

FIG. 5e

09687540-101300







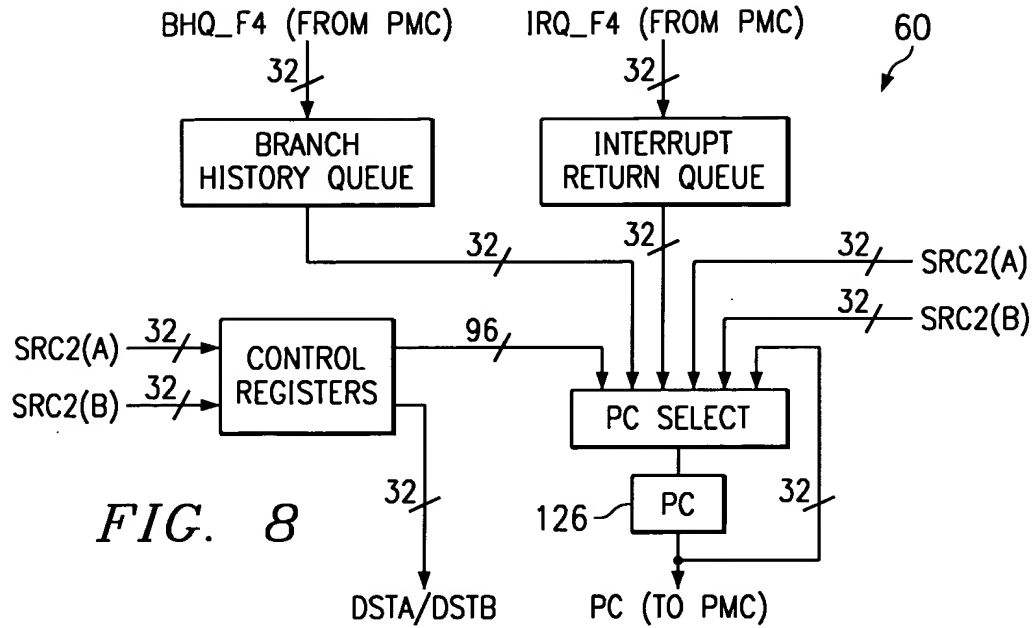


FIG. 8

|| [PREDICATION REG] INSTRUCTION\_MNEMONIC .UNIT-DATAPATH-CROSSPATH OP1, OP2, DST

WHERE:

|| =TO BE SCHEDULED IN PARALLEL WITH PRECEDING INSTRUCTION(S)  
 [PREDICATION REG] =REGISTER CONTAINING PREDICATION VALUE  
 .UNIT =A,C,S,M,D,P UNIT GROUPS  
 DATAPATH =1 FOR DATAPATH A, 2 FOR DATAPATH B  
 CROSSPATH =X IF ONE OPERAND COMES FROM OPPOSITE REGISTER FILE  
 OP1, OP2 =SOURCE REGISTERS  
 DST =DESTINATION REGISTER

UNIT GROUP	ASSEMBLY NOTATIONS		ASSEMBLY EXAMPLES	WITH CROSSPATH
	DATAPATH A	DATAPATH B		
A	.A1	.A2	ADD .A1 A1,A2,A3 SUB .A2 B1,B2,B3	ADD .A1X A1,B2,A3 SUB .A2X B1,A2,B3
C	.C1	.C2	CMPEQ .C1 A1,A2,A3 CMPEQ .C2 B1,B2,B3	CMPEQ .C1X A1,B2,A3 CMPEQ .C2X B1,A2,B3
S	.S1	.S2	SHL .S1 A1,A2,A3 SHL .S2 B1,B2,B3	SHL .S1X A1,B2,A3 SHL .S2X B1,A2,B3
M	.M1	.M2	MPY .M1 A1,A2,A3 MPY .M2 B1,B2,B3	MPY .M1X A1,B2,A3 MPY .M2X B1,A2,B3
D	.D		LDB .D *A8,A12 STB .D A8,*A12 ADDAH .D A8,A2,B1	n/a
P	.P		B A8	n/a

FIG. 15



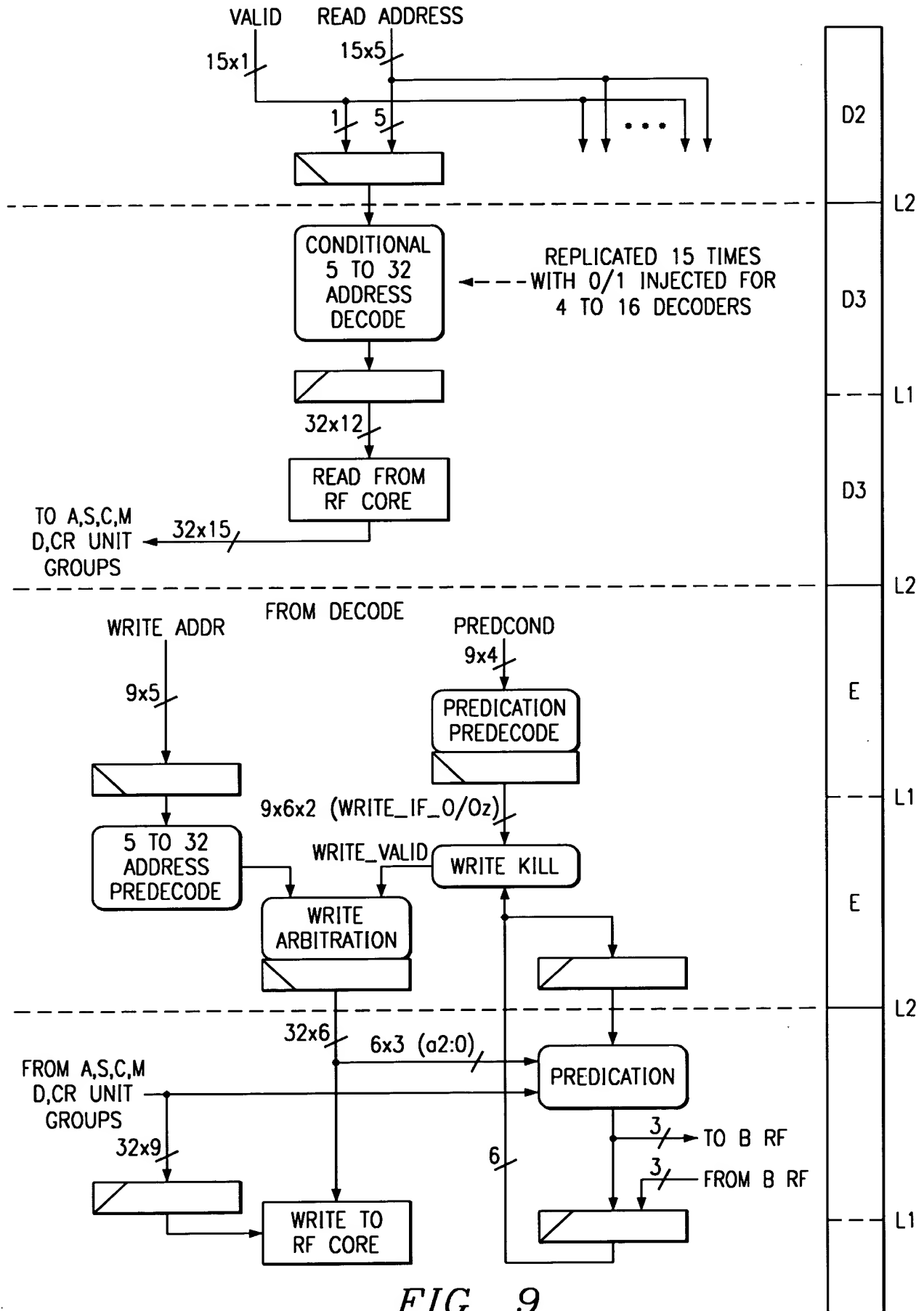


FIG. 9

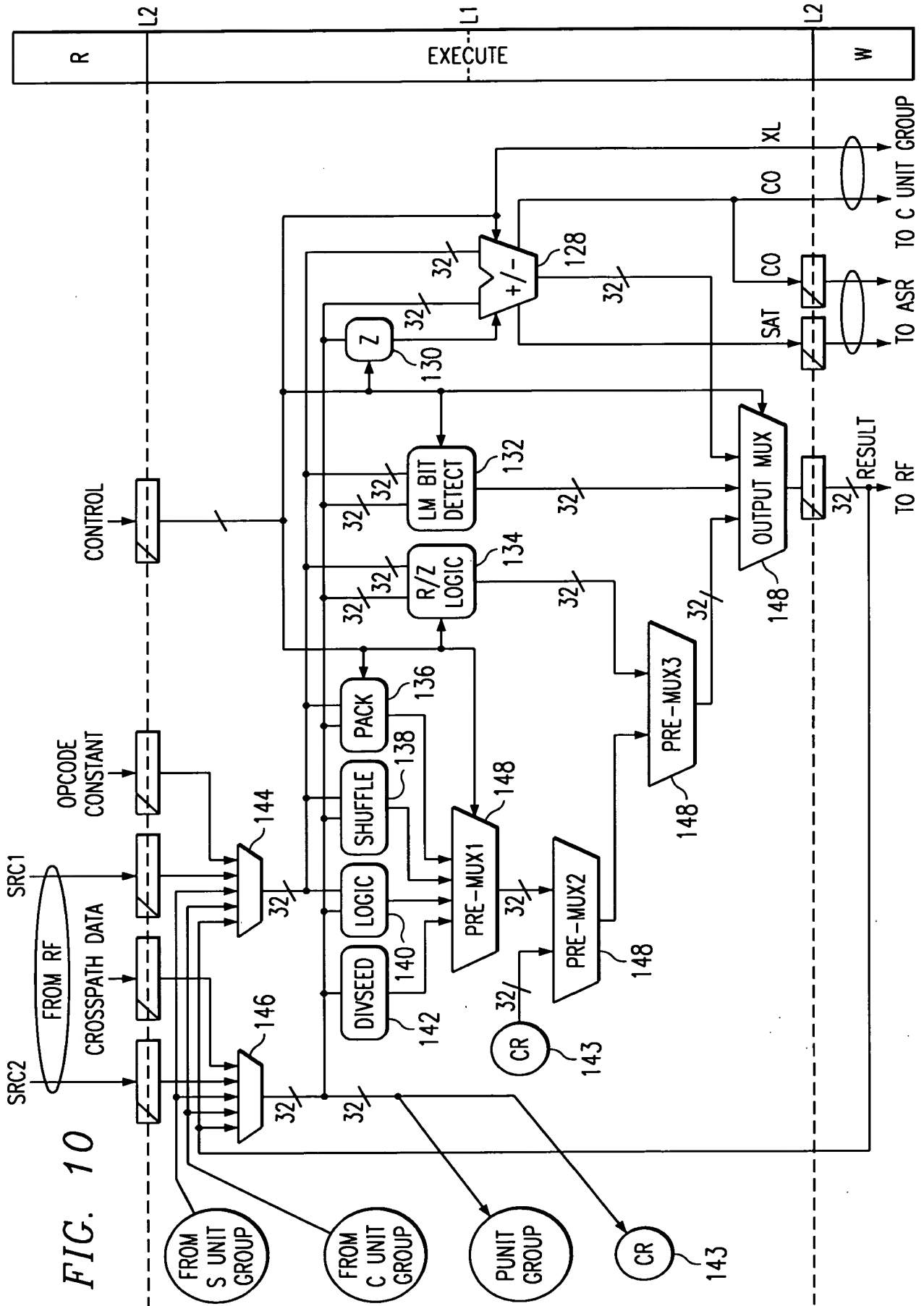
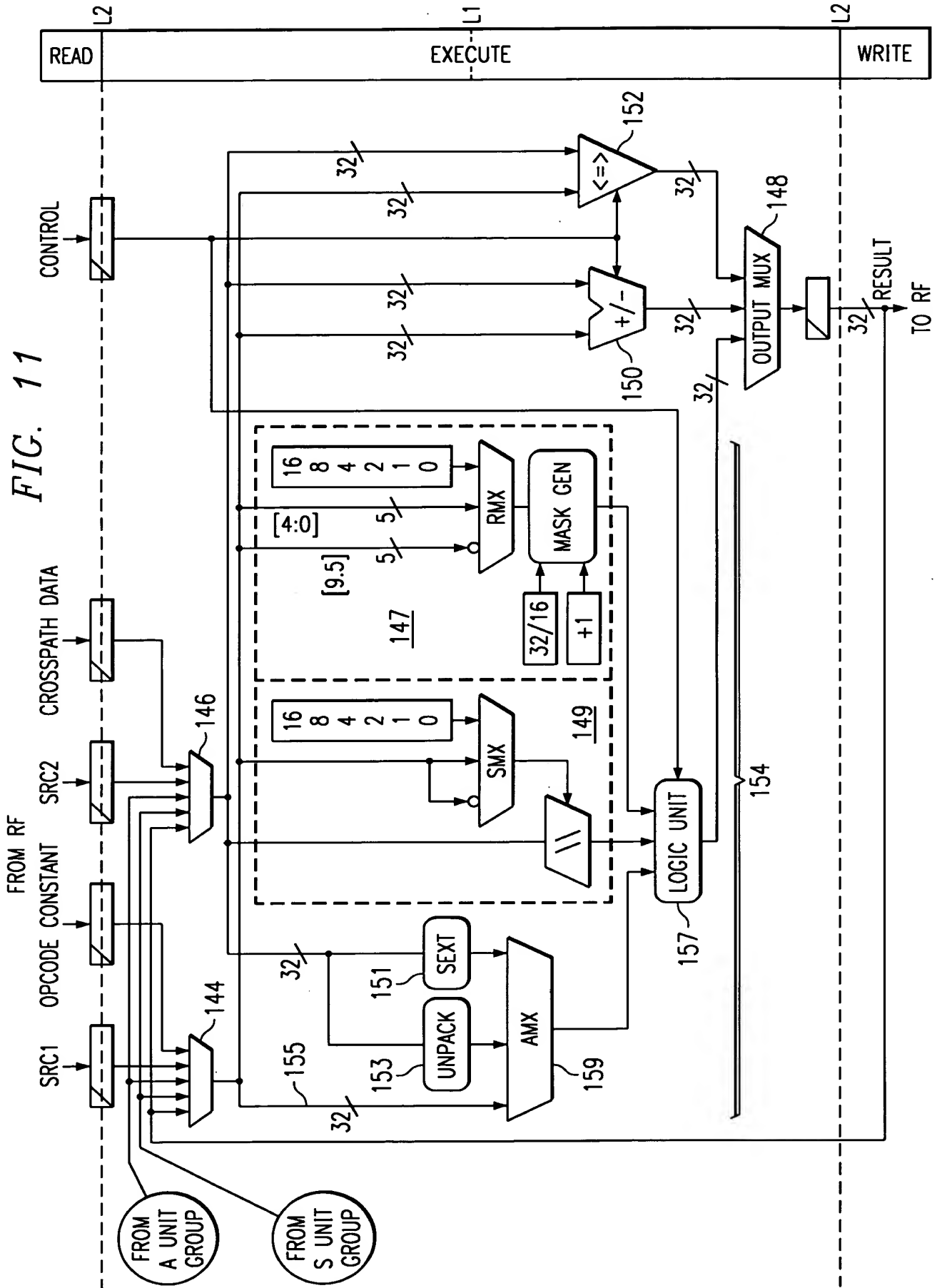
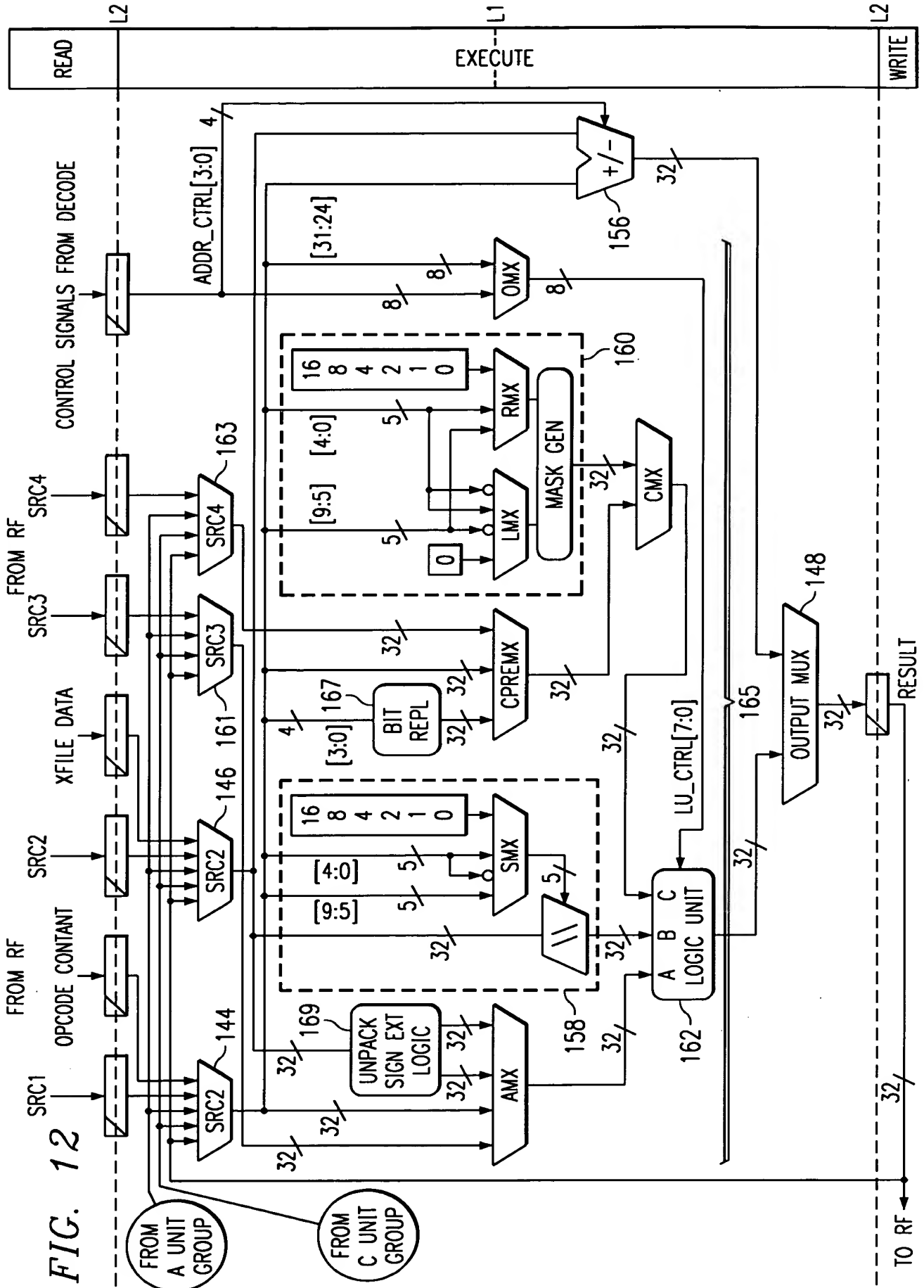
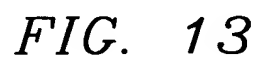


FIG. 11







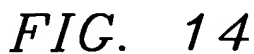


FIG. 16

Mnemonic	Action	Operation
LDB[U] <u>168</u>	Load byte	Memory B/A XXXXXXba → SSSSSba (signed (s)) XXXXXXba → 00000ba (unsigned (u))
LDH[U] <u>170</u>	Load halfword	Memory B/A XXXXdcba → SSSSdcba (s) XXXXdcba → 0000dcba (u)
LDW <u>172</u>	Load word	Memory B/A hgfedcba → hgfedcba
LDD <u>174</u>	Load double	Memory B/O A/E ponmlkji hgfedcba → ponmlkji hgfedcba

FIG. 17

Mnemonic	Action	Operation
STB <u>176</u>	Store byte	B/A Memory XXXXXXba → 000000ba
STH <u>178</u>	Store halfword	B/A Memory XXXXdcba → 0000dcba
STW <u>180</u>	Store word	B/A Memory hgfedcba → hgfedcba
STD <u>182</u>	Store double	B/O A/E Memory ponmlkji hgfedcba → ponmlkji hgfedcba

Mnemonic	Action	Operation
LDW_BH[U] <u>184</u>	Word: unpack the bytes into halfwords	Memory      B/O      A/E hgfedcba → SShgSSfe SSdcSSba      (signed (s)) hgfedcba → 00hg00fe 00dc00ba      (unsigned (u))
LDW_BHI[U] <u>186</u>	Word: unpack the bytes into halfwords interleaved	Memory      B/O      A/E hgfedcba → SShgSSdc SSfeSSba      (s) hgfedcba → 00hg00dc 00fe00ba      (u)
LDW_HW[U] <u>188</u>	Word: unpack the halfwords into words	Memory      B/O      A/E hgfedcba → SSSShgfe SSSSdcba      (s) hgfedcba → 0000hgfe 0000dcba      (u)
LDD_BH[U] <u>190</u>	Double: unpack the bytes into halfwords	Memory      BO      BE      AO      AE ponmlkji hgfedcba → SSpoSSnm SSikSSji      SSdcSSba      (s) ponmlkji hgfedcba → 00po00nm 00lk00ji      00hg00fe      00dc00ba      (u)
LDD_BHI[U] <u>192</u>	Double: unpack the bytes into halfwords interleaved	Memory      BO      BE      AO      AE ponmlkji hgfedcba → SSpoSSik SSnmSSji      SSdcSSba      (s) ponmlkji hgfedcba → 00po00lk 00nm00ji      00hg00dc      00fe00ba      (u)
LDD_HW[U] <u>194</u>	Double: unpack the halfwords into words	Memory      BO      BE      AO      AE ponmlkji hgfedcba → SSSSponm SSSSlkji      SSSShgfe      SSSSdcba      (s) ponmlkji hgfedcba → 0000ponm 0000lkji      0000hgfe      0000dcba      (u)
LDD_HWI[U] <u>196</u>	Double: unpack the halfwords into words interleaved	Memory      BO      BE      AO      AE ponmlkji hgfedcba → SSSSponm SSSShgfe      SSSSlkji      SSSSdcba      (s) ponmlkji hgfedcba → 0000ponm 0000hgfe      0000lkji      0000dcba      (u)

FIG. 18



Mnemonic	Action	Operation
STBH_W <u>198</u>	Pack the LS byte of each halfword into a word	B/O A/E Memory XXhgXXfe XXdcXXba → hgfedcba
STBHI_W <u>200</u>	Pack the LS byte of each halfword interleaved into a word	B/O A/E Memory XXhgXXdc XXfeXXba → hgfedcba
STHW_W <u>202</u>	Pack the LS halfword of each word into a word	B/O A/E Memory XXXXhgfe XXXXdcb → hgfedcba
STBH_D <u>204</u>	Pack the LS byte of each halfword into a double	B/O BE AO AE Memory XXpoXXnm XXlkXXji XXhgXXfe XXdcXXba → ponmlkji hgfedcba
STBHI_D <u>206</u>	Pack the LS byte of each halfword interleaved into a double	B/O BE AO AE Memory XXpoXXlk XXnmXXji XXhgXXdc XXfeXXba → ponmlkji hgfedcba
STHW_D <u>208</u>	Pack the LS halfword of each word into a double	B/O BE AO AE Memory XXXXponm XXXXlkji XXXXhgfe XXXXdcb → ponmlkji hgfedcba
STHWI_D <u>210</u>	Pack the LS halfword of each word interleaved into a double	B/O BE AO AE Memory XXXXponm XXXXhgfe XXXXlkji XXXXdcb → ponmlkji hgfedcba

FIG. 19